

Four-Dimensional Gallant-Lambert-Vanstone Scalar Multiplication

Peter Birkner¹, Patrick Longa², and Francesco Sica³

¹ Bundesamt für Sicherheit in der Informationstechnik (BSI),
Godesberger Allee 185-189, 53175 Bonn, Germany

`peter.birkner@bsi.bund.de`

² Microsoft Research,
One Microsoft Way, Redmond, WA 98052, USA

`plonga@microsoft.com`

³ Via Toscana 50, Prata (GR), 58024 Italy
`fracrypto@gmail.com`

Abstract. The GLV method of Gallant, Lambert and Vanstone (CRYPTO 2001) computes any multiple kP of a point P of prime order n lying on an elliptic curve with a low-degree endomorphism Φ (called GLV curve) over \mathbb{F}_p as

$$kP = k_1P + k_2\Phi(P), \quad \text{with } \max\{|k_1|, |k_2|\} \leq C_1\sqrt{n}$$

for some explicit constant $C_1 > 0$. Recently, Galbraith, Lin and Scott (EUROCRYPT 2009) extended this method to all curves over \mathbb{F}_{p^2} which are twists of curves defined over \mathbb{F}_p . We show in this work how to merge the two approaches in order to get, for twists of any GLV curve over \mathbb{F}_{p^2} , a four-dimensional decomposition together with fast endomorphisms Φ, Ψ over \mathbb{F}_{p^2} acting on the group generated by a point P of prime order n , resulting in a proved decomposition for any scalar $k \in [1, n]$

$$kP = k_1P + k_2\Phi(P) + k_3\Psi(P) + k_4\Psi\Phi(P) \quad \text{with } \max_i(|k_i|) < C_2 n^{1/4}$$

for some explicit $C_2 > 0$. Furthermore, taking the best C_1, C_2 , we get $C_2/C_1 < 408$, independently of the curve, ensuring a constant relative speedup.

We also derive new families of GLV curves, corresponding to those curves with degree 3 endomorphisms.

Keywords. Elliptic curves, GLV method, Scalar Multiplication.

1 Introduction

The Gallant-Lambert-Vanstone (GLV) method is a generic method to speed up computation on some elliptic curves over fields of large characteristic. Given a curve with a point P of prime order n , it consists essentially in an algorithm to find a decomposition of an arbitrary scalar multiplication kP for $k \in [1, n]$ into two scalar multiplications with the new scalars having only about half the original bits. We call such a method two-dimensional, since if scalar multiplications can be parallelized, then a twofold performance speedup can be achieved.

Whereas the original GLV method as defined in [3] works on curves over \mathbb{F}_p with an endomorphism of small degree (GLV curves), Galbraith-Lin-Scott (GLS) in [2] have shown that over \mathbb{F}_{p^2} one can expect to find many more such curves by basically exploiting the action of the Frobenius endomorphism. One can therefore expect that on the particular GLV curves, this new insight will lead to improvements over \mathbb{F}_{p^2} . Indeed the GLS article itself considers fourfold speedups on GLV curves with nontrivial automorphisms (corresponding to the degree one cases) but leaves the other cases open to investigation.

Recently a paper by Zhou, Hu, Xu and Song [12] has shown that it is possible to combine the two approaches by introducing a three-dimensional version of the GLV method (thus getting three scalars with a threefold speedup), which seems to be working to a certain degree, with however no justification but through practical implementations.

In contrast, we would like to show that the most natural understanding of their ideas is in four dimensions, where we are then able to construct, for the same curves and fast endomorphisms Φ, Ψ over \mathbb{F}_{p^2} acting on a cyclic group generated by a point P of prime order n , a proved decomposition for any scalar $k \in [1, n]$

$$kP = k_1P + k_2\Phi(P) + k_3\Psi(P) + k_4\Psi\Phi(P) \quad \text{with } \max_i(|k_i|) < Cn^{1/4}$$

for some explicitly computable C . If parallel computation is available, then the computation of kP can possibly be implemented up to four times as fast as a traditional scalar multiplication. It recently came to our attention that Hu, Longa and Xu [4] also provided a similar bound in the case of curves with j -invariant 0. Our analysis supplements theirs by considering all GLV curves, where we provide an unified treatment.

The way to prove this bound is to study the kernel lattice of the GLV reduction map in dimension four. The LLL algorithm [5] then will find a suitable reduced basis together with a useful bound to deduce C . In the last part of the article, we develop another approach which gives a reduced basis faster than the LLL algorithm together with a much better value for C . Indeed our reduction algorithm runs in $O(\log^2 n)$ compared to $O(\log^3 n)$ for LLL and the improved $C = O(\sqrt{s})$ compared to the value obtained with LLL which is only $\Omega(s^{3/2})$. This allows us to prove that the relative speedup in going from a two-dimensional to a four-dimensional GLV method is independent of the curve.

2 The GLV Method

In this section we briefly summarize the GLV method following [9]. Let E be an elliptic curve defined over a finite field \mathbb{F}_q and P be a point of this curve with prime order n such that the cofactor $h = \#E(\mathbb{F}_q)/n$ is small, say $h \leq 4$. Let us consider Φ a non trivial endomorphism defined over \mathbb{F}_q and $X^2 + rX + s$ its characteristic polynomial. In all the examples r and s are actually small fixed integers and q is varying in some family. By hypothesis there is only one subgroup of order n in $E(\mathbb{F}_q)$, implying that $\Phi(P) = \lambda P$ for some $\lambda \in [0, n-1]$, since $\Phi(P)$ has order dividing the prime n . In particular, λ is obtained as a root of $X^2 + rX + s$ modulo n .

Define the group homomorphism (the GLV reduction map)

$$\begin{aligned} \mathfrak{f}: \mathbb{Z} \times \mathbb{Z} &\rightarrow \mathbb{Z}/n \\ (i, j) &\mapsto i + \lambda j \pmod{n} . \end{aligned}$$

Let $\mathcal{K} = \ker \mathfrak{f}$. It is a sublattice of $\mathbb{Z} \times \mathbb{Z}$ of rank 2 since the quotient is finite. Let $\mathbb{k} > 0$ be a constant (depending on the curve) such that we can find v_1, v_2 two linearly independent vectors of \mathcal{K} satisfying $\max\{|v_1|, |v_2|\} < \mathbb{k}\sqrt{n}$, where $|\cdot|$ denotes the rectangle norm⁴. Express

$$(k, 0) = \beta_1 v_1 + \beta_2 v_2 ,$$

⁴ The rectangle norm of (x, y) is by definition $\max(|x|, |y|)$. As remarked in [9], we can replace it by any other metric norm. We will use the term "short" to denote smallness in the rectangle norm.

where $\beta_i \in \mathbb{Q}$. Then round β_i to the nearest integer $b_i = \lfloor \beta_i \rfloor = \lfloor \beta_i + 1/2 \rfloor$ and let $v = b_1 v_1 + b_2 v_2$. Note that $v \in \mathcal{K}$ and that $u \stackrel{\text{def}}{=} (k, 0) - v$ is short. Indeed by the triangle inequality we have that

$$|u| \leq \frac{|v_1| + |v_2|}{2} < \mathbb{k} \sqrt{n}.$$

If we set $(k_1, k_2) = u$, then we get $k \equiv k_1 + k_2 \lambda \pmod{n}$ or equivalently $kP = k_1 P + k_2 \Phi(P)$, with $\max(|k_1|, |k_2|) < \mathbb{k} \sqrt{n}$.

In [9], the optimal value of \mathbb{k} (with respect to large values of n , i.e. large fields, keeping $X^2 + rX + s$ constant) is determined. Let $\Delta = r^2 - 4s$ be the discriminant of the characteristic polynomial of Φ . Then the optimal \mathbb{k} is given by the following result⁵.

Theorem 1 ([9, Theorem 4]). *Assuming n is the norm of an element of $\mathbb{Z}[\Phi]$, then the optimal value of \mathbb{k} is*

$$\mathbb{k} = \begin{cases} \frac{\sqrt{s}}{2} \left(1 + \frac{1}{|\Delta|}\right), & \text{if } r \text{ is odd,} \\ \frac{\sqrt{s}}{2} \sqrt{1 + \frac{4}{|\Delta|}}, & \text{if } r \text{ is even.} \end{cases}$$

3 The GLS Improvement

In 2009, Galbraith, Lin and Scott [2] realised that we don't need to have $\Phi^2 + r\Phi + s = 0$ in $\text{End}(E)$ but only in a subgroup of $E(\mathbb{F})$ for a specific finite field \mathbb{F} . In particular, considering $\Psi = \text{Frob}_p$ the p -Frobenius endomorphism of a curve E defined over \mathbb{F}_p , we know that $\Psi^m(P) = P$ for all $P \in E(\mathbb{F}_{p^m})$. While this says nothing useful if $m = 1, 2$, it does offer new nontrivial relations for higher degree extensions. The case $m = 4$ is particularly useful here.

In this case if $P \in E(\mathbb{F}_{p^4}) \setminus E(\mathbb{F}_{p^2})$, then $\Psi^2(P) = -P$ and hence on the subgroup generated by P , Ψ satisfies the equation $X^2 + 1 = 0$. This implies that if $\Psi(P)$ is a multiple of P (which happens as soon as the order n of P is sufficiently large, say at least $2p$), we can apply the previous GLV construction and split again a scalar multiplication as $kP = k_1 P + k_2 \Psi(P)$, with $\max(|k_1|, |k_2|) = O(\sqrt{n})$. Contrast this with the characteristic polynomial of Ψ which is $X^2 - a_p X + p$ for some integer a_p , a non-constant polynomial to which we cannot apply as efficiently the GLV paradigm.

For efficiency reasons however one does not work with E/\mathbb{F}_{p^4} directly but with E'/\mathbb{F}_{p^2} isomorphic to E over \mathbb{F}_{p^4} but not over \mathbb{F}_{p^2} , that is, a quadratic twist over \mathbb{F}_{p^2} . In this case, it's possible that $\#E'(\mathbb{F}_{p^2}) = n \geq (p-1)^2$ be prime. Furthermore, if $\psi: E' \rightarrow E$ is an isomorphism defined over \mathbb{F}_{p^4} , then the endomorphism $\Psi = \psi \text{Frob}_p \psi^{-1} \in \text{End}(E')$ satisfies the equation $X^2 + 1 = 0$ and if $p \equiv 5 \pmod{8}$ it can be defined over \mathbb{F}_p .

This idea is at the heart of the GLS approach, but it only works for curves over \mathbb{F}_{p^m} with $m > 1$, therefore it does not generalise the original GLV method but rather complements it.

⁵ There is a mistake in [9] in the derivation of \mathbb{k} for odd values of r . This affects [9, Corollary 1] for curves E_2 and E_3 , where the correct values of \mathbb{k} are respectively $2/3$ and $4\sqrt{2}/7$.

4 Examples

We give a few examples of GLV curves, which are curves defined over \mathbb{C} with complex multiplication by an quadratic integer of small norm, corresponding to an endomorphism ϕ of small degree⁶. They make up an exhaustive list, up to isomorphism, in increasing order of endomorphism degree up to degree 3. While the first four examples appear in the previous literature, the next ones (degree 3) are new and have been computed with the Stark algorithm [10].

Example 1. Let $p \equiv 1 \pmod{4}$ be a prime. Define an elliptic curve E_1 over \mathbb{F}_p by

$$y^2 = x^3 + ax \ .$$

If β is an element of order 4, then the map ϕ defined in the affine plane by

$$\phi(x, y) = (-x, \beta y) \ ,$$

is an endomorphism of E_1 defined over \mathbb{F}_p with $\text{End}(E_1) = \mathbb{Z}[\phi] \cong \mathbb{Z}[\sqrt{-1}]$, since ϕ satisfies the equation

$$\phi^2 + 1 = 0 \ .$$

Example 2. Let $p \equiv 1 \pmod{3}$ be a prime. Define an elliptic curve E_2 over \mathbb{F}_p by

$$y^2 = x^3 + b \ .$$

If γ is an element of order 3, then we have an endomorphism ϕ defined over \mathbb{F}_p by

$$\phi(x, y) = (\gamma x, y) \ ,$$

and $\text{End}(E_2) = \mathbb{Z}[\phi] \cong \mathbb{Z}[\frac{1+\sqrt{-3}}{2}]$, since ϕ satisfies the equation

$$\phi^2 + \phi + 1 = 0 \ .$$

Example 3. Let $p > 3$ be a prime such that -7 is a quadratic residue modulo p . Define an elliptic curve E_3 over \mathbb{F}_p by

$$y^2 = x^3 - \frac{3}{4}x^2 - 2x - 1 \ .$$

If $\xi = (1 + \sqrt{-7})/2$ and $a = (\xi - 3)/4$, then we get the \mathbb{F}_p -endomorphism ϕ defined by

$$\phi(x, y) = \left(\frac{x^2 - \xi}{\xi^2(x - a)}, \frac{y(x^2 - 2ax + \xi)}{\xi^3(x - a)^2} \right) \ ,$$

and $\text{End}(E_3) = \mathbb{Z}[\phi] \cong \mathbb{Z}[\frac{1+\sqrt{-7}}{2}]$, since ϕ satisfies the equation

$$\phi^2 - \phi + 2 = 0 \ .$$

⁶ By small we mean really small, usually less than 5. In particular, for cryptographic applications, the degree is much smaller than the field size.

Example 4. Let $p > 3$ be a prime such that -2 is a quadratic residue modulo p . Define an elliptic curve E_4 over \mathbb{F}_p by

$$y^2 = 4x^3 - 30x - 28$$

together with the \mathbb{F}_p -endomorphism ϕ defined⁷ by

$$\phi(x, y) = \left(-\frac{2x^2 + 4x + 9}{4(x+2)}, y \frac{2x^2 + 8x - 1}{4\sqrt{-2}(x+2)^2} \right).$$

We have $\text{End}(E_4) = \mathbb{Z}[\phi] \cong \mathbb{Z}[\sqrt{-2}]$ since ϕ satisfies the equation

$$\phi^2 + 2 = 0.$$

Example 5. Let $p > 3$ be a prime such that -11 is a quadratic residue mod p . We define the elliptic curve E_5 over \mathbb{F}_p

$$y^2 = x^3 - \frac{13824}{539}x + \frac{27648}{539}$$

with $a = (1 + \sqrt{-11})/2$ and the endomorphism ϕ defined by

$$\begin{aligned} \phi(x, y) = & \left(\frac{\left(-\frac{539}{5184}a + \frac{539}{1728} \right) x^3 + \left(\frac{28}{27}a - \frac{35}{18} \right) x^2 + \left(-\frac{92}{9}a + \frac{8}{3} \right) x + \frac{1728}{77}a + \frac{192}{77}}{\left(\frac{2695}{5184}a - \frac{539}{864} \right) x^2 + \left(-\frac{217}{54}a + \frac{49}{18} \right) x + \frac{64}{9}a - \frac{4}{3}}, y \right. \\ & \left. \frac{\left(\frac{3773}{373248}a - \frac{18865}{995328} \right) x^3 + \left(-\frac{2695}{20736}a + \frac{539}{3456} \right) x^2 + \left(\frac{7}{432}a - \frac{91}{144} \right) x + \frac{20}{27}a + \frac{1}{9}}{\left(-\frac{18865}{1492992}a + \frac{116963}{995328} \right) x^3 + \left(\frac{7007}{20736}a - \frac{539}{432} \right) x^2 + \left(-\frac{791}{432}a + \frac{581}{144} \right) x + \frac{74}{27}a - \frac{35}{9}} \right) \end{aligned}$$

such that $\text{End}(E_5) = \mathbb{Z}[\phi] \cong \mathbb{Z}[\frac{1+\sqrt{-11}}{2}]$. The characteristic polynomial of ϕ is

$$\phi^2 - \phi + 3 = 0.$$

Example 6. Let $p > 3$ be a prime such that -3 is a quadratic residue mod p . We define the elliptic curve E_6 over \mathbb{F}_p

$$y^2 = x^3 - \frac{3375}{121}x + \frac{6750}{121}$$

with the endomorphism ϕ defined by

$$\begin{aligned} \phi(x, y) = & \left(-\frac{1331x^3 - 10890x^2 + 81675x - 189000}{33(11x - 45)^2}, \right. \\ & \left. y \frac{1331x^3 - 16335x^2 + 7425x + 43875}{3\sqrt{-3}(11x - 45)^3} \right) \end{aligned}$$

such that⁸ $\text{End}(E_6) = \mathbb{Z}[\phi] \cong \mathbb{Z}[\sqrt{-3}]$. The characteristic polynomial of ϕ is

$$\phi^2 + 3 = 0.$$

⁷ We take the opportunity to correct a typo found and transmitted in many sources, where a y factor was absent in the second coordinate. Its sign is irrelevant.

⁸ This is the first example where the endomorphism ring is not the maximal order of its field of fractions. It can be summarily seen as follows: $\text{End}(E) \supseteq \mathbb{Z}[\sqrt{-3}]$. If not equal, then it must be the full ring of integers $\mathbb{Z}[\frac{1+\sqrt{-3}}{2}]$. This would imply that $j = 0$, as there is only $h(-3) = 1$ isomorphism class of elliptic curves with complex multiplication by $\mathbb{Z}[\frac{1+\sqrt{-3}}{2}]$, given in Example 2 (see [10] for an abridged description of the theory of complex multiplication). This is clearly not the case here. Alternatively, one can see that there would exist a nontrivial automorphism (a primitive cube root of unity) corresponding to $\frac{-1+\sqrt{-3}}{2}$. A direct computation then shows this is impossible.

5 Combining GLV and GLS

Let E/\mathbb{F}_p be a GLV curve. As in Section 3, we will denote by E'/\mathbb{F}_{p^2} a quadratic twist \mathbb{F}_{p^4} -isomorphic to E via the isomorphism $\psi: E' \rightarrow E$. We also suppose that $\#E'(\mathbb{F}_{p^2}) = nh$ where n is prime and $h \leq 4$. We then have the two endomorphisms of E' , $\Psi = \psi \text{Frob}_p \psi^{-1}$ and $\Phi = \psi \phi \psi^{-1}$, with ϕ the GLV endomorphism coming with the definition of a GLV curve. They are both defined over \mathbb{F}_{p^2} , since if σ is the nontrivial Galois automorphism of $\mathbb{F}_{p^4}/\mathbb{F}_{p^2}$, then $\psi^\sigma = -\psi$, so that $\Psi^\sigma = \psi^\sigma \text{Frob}_p^\sigma (\psi^{-1})^\sigma = (-\psi) \text{Frob}_p(-\psi^{-1}) = \Psi$, meaning that $\Psi \in \text{End}_{\mathbb{F}_{p^2}}(E')$. Similarly for Φ , where we are using the fact that $\phi \in \text{End}_{\mathbb{F}_p}(E)$. Notice that $\Psi^2 + 1 = 0$ and that Φ has the same characteristic polynomial as ϕ . Furthermore, since we have a large subgroup $\langle P \rangle \subset E'(\mathbb{F}_{p^2})$ of prime order, $\Phi(P) = \lambda P$ and $\Psi(P) = \mu P$ for some $\lambda, \mu \in [1, n-1]$. We will assume that Φ and Ψ , when viewed as algebraic integers, generate disjoint quadratic extensions of \mathbb{Q} . In particular, we are not dealing with Example 1, but this can be treated separately with a quartic twist, as was hinted in [2].

Consider the biquadratic (Galois of degree 4, with Galois group $\mathbb{Z}/2 \times \mathbb{Z}/2$) number field $K = \mathbb{Q}(\Phi, \Psi)$. Let \mathfrak{o}_K be its ring of integers. The following analysis is inspired by Sica, Ciet and Quisquater [9, Section 8].

We have $\mathbb{Z}[\Phi, \Psi] \subseteq \mathfrak{o}_K$. Since the degrees of Φ and Ψ are much smaller when compared to n , the prime n is unramified in K and the existence of λ and μ above means that n splits in $\mathbb{Q}(\Phi)$ and $\mathbb{Q}(\Psi)$, namely that n splits completely in K . There exists therefore a prime ideal \mathfrak{n} of \mathfrak{o}_K dividing $n\mathfrak{o}_K$, such that its norm is n . We can also suppose that $\Phi \equiv \lambda \pmod{\mathfrak{n}}$ and $\Psi \equiv \mu \pmod{\mathfrak{n}}$. The four-dimensional GLV (4-GLV) method works as follows.

Consider the 4-GLV reduction map F defined by

$$\begin{aligned} F: \mathbb{Z}^4 &\rightarrow \mathbb{Z}/n \\ (x_1, x_2, x_3, x_4) &\mapsto x_1 + x_2\lambda + x_3\mu + x_4\lambda\mu \pmod{n} . \end{aligned}$$

If we can find four linearly independent vectors $v_1, \dots, v_4 \in \ker F$, with $\max_i |v_i| \leq Cn^{1/4}$ for some constant $C > 0$, then for any $k \in [1, n-1]$ we write

$$(k, 0, 0, 0) = \sum_{j=1}^4 \beta_j v_j ,$$

with $\beta_j \in \mathbb{Q}$. As in the GLV method one sets $v = \sum_{j=1}^4 \lfloor \beta_j \rfloor v_j$ and

$$u = (k, 0, 0, 0) - v = (k_1, k_2, k_3, k_4) .$$

We then get

$$kP = k_1P + k_2\Phi(P) + k_3\Psi(P) + k_4\Psi\Phi(P) \quad \text{with} \quad \max_i (|k_i|) \leq 2Cn^{1/4} . \quad (1)$$

We focus next on the study of $\ker F$ in order to find a reduced basis v_1, v_2, v_3, v_4 with an explicit C . We can factor the 4-GLV map F as

$$\begin{array}{ccc} \mathbb{Z}^4 & \xrightarrow{f} & \mathbb{Z}[\Phi, \Psi] \xrightarrow[\text{mod } \mathfrak{n} \cap \mathbb{Z}[\Phi, \Psi]]{\text{reduction}} \mathbb{Z}/n \\ (x_1, x_2, x_3, x_4) & \mapsto & x_1 + x_2\Phi + x_3\Psi + x_4\Phi\Psi \mapsto x_1 + x_2\lambda + x_3\mu + x_4\lambda\mu \\ & & \pmod{n} . \end{array}$$

Notice that the kernel of the second map (reduction mod $\mathfrak{n} \cap \mathbb{Z}[\Phi, \Psi]$) is exactly $\mathfrak{n} \cap \mathbb{Z}[\Phi, \Psi]$. This can be seen as follows. The reduction map factors as

$$\mathbb{Z}[\Phi, \Psi] \longrightarrow \mathfrak{o}_K \longrightarrow \mathfrak{o}_K/\mathfrak{n} \cong \mathbb{Z}/n$$

where the first arrow is inclusion, the second is reduction mod \mathfrak{n} , corresponding to reducing the x_i 's mod $\mathfrak{n} \cap \mathbb{Z} = n\mathbb{Z}$ and using $\Phi \equiv \lambda, \Psi \equiv \mu \pmod{\mathfrak{n}}$. But the kernel of this map consists precisely of elements of $\mathbb{Z}[\Phi, \Psi]$ which are in \mathfrak{n} , and that is what we want.

Moreover, since the reduction map is surjective, we obtain an isomorphism $\mathbb{Z}[\Phi, \Psi]/\mathfrak{n} \cap \mathbb{Z}[\Phi, \Psi] \cong \mathbb{Z}/n$ which says that the index of $\mathfrak{n} \cap \mathbb{Z}[\Phi, \Psi]$ inside $\mathbb{Z}[\Phi, \Psi]$ is n . Since the first map f is an isomorphism, we get that $\ker F = f^{-1}(\mathfrak{n} \cap \mathbb{Z}[\Phi, \Psi])$ and that $\ker F$ has index $[\mathbb{Z}^4 : \ker F] = n$ inside \mathbb{Z}^4 .

We can also produce a basis of $\ker F$ by the following observation. Let $\Phi' = \Phi - \lambda, \Psi' = \Psi - \mu$, hence $\Phi'\Psi' = \Phi\Psi - \lambda\Psi - \mu\Phi + \lambda\mu$. In matrix form,

$$\begin{pmatrix} 1 \\ \Phi' \\ \Psi' \\ \Phi'\Psi' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\lambda & 1 & 0 & 0 \\ -\mu & 0 & 1 & 0 \\ \lambda\mu & -\mu & -\lambda & 1 \end{pmatrix} \begin{pmatrix} 1 \\ \Phi \\ \Psi \\ \Phi\Psi \end{pmatrix}$$

Since the determinant of the square matrix is 1, we deduce that $\mathbb{Z}[\Phi, \Psi] = \mathbb{Z}[\Phi', \Psi']$. But in this new basis, we claim that

$$\mathfrak{n} \cap \mathbb{Z}[\Phi', \Psi'] = n\mathbb{Z} + \mathbb{Z}\Phi' + \mathbb{Z}\Psi' + \mathbb{Z}\Phi'\Psi'.$$

Indeed, reverse inclusion (\supseteq) is easy since $\Phi', \Psi', \Phi'\Psi' \in \mathfrak{n}$ and so is n , because \mathfrak{n} divides $n\mathfrak{o}_K$ is equivalent to $\mathfrak{n} \supseteq n\mathfrak{o}_K$. On the other hand, the index of both sides in $\mathbb{Z}[\Phi', \Psi']$ is n , which can only happen, once an inclusion is proved, if the two sides are equal. Using the isomorphism f , we see that a basis of $\ker F \subset \mathbb{Z}^4$ is therefore given by

$$w_1 = (n, 0, 0, 0), w_2 = (-\lambda, 1, 0, 0), w_3 = (-\mu, 0, 1, 0), w_4 = (\lambda\mu, -\mu, -\lambda, 1).$$

The LLL algorithm [5] then finds, for a given basis w_1, \dots, w_4 of $\ker F$, a reduced⁹ basis v_1, \dots, v_4 in polynomial time (in the logarithm of the norm of the w_i 's) such that (cf. [1, Theorem 2.6.2 p.85])

$$\prod_{i=1}^4 |v_i| \leq 8 [\mathbb{Z}^4 : \ker F] = 8n. \quad (2)$$

Lemma 1. *Let*

$$\mathcal{N}: \mathbb{Z}^4 \rightarrow \mathbb{Z} \\ (x_1, x_2, x_3, x_4) \mapsto \sum_{\substack{i_1, i_2, i_3, i_4 \geq 0 \\ i_1 + i_2 + i_3 + i_4 = 4}} b_{i_1, i_2, i_3, i_4} x_1^{i_1} x_2^{i_2} x_3^{i_3} x_4^{i_4}$$

be the norm of an element $x_1 + x_2\Phi + x_3\Psi + x_4\Phi\Psi \in \mathbb{Z}[\Phi, \Psi]$, where the b_{i_1, i_2, i_3, i_4} 's lie in \mathbb{Z} . Then, for any nonzero $v \in \ker F$, one has

$$|v| \geq \frac{n^{1/4}}{\left(\sum_{\substack{i_1, i_2, i_3, i_4 \\ i_1 + i_2 + i_3 + i_4 = 4}} |b_{i_1, i_2, i_3, i_4}| \right)^{1/4}}. \quad (3)$$

⁹ The estimates are usually given for the Euclidean norm of the vectors. But it is easy to see that the rectangle norm is upper bounded by the Euclidean norm.

Proof. For $v \in \ker F$ we have $\mathcal{N}(v) \equiv 0 \pmod{n}$ and if $v \neq 0$ we must therefore have $|\mathcal{N}(v)| \geq n$. On the other hand, if we did not have (3), then every component of v would be strictly less than the right-hand side and plugging this upper bound in the definition of $|\mathcal{N}(v)|$ would yield a quantity $< n$, a contradiction. \square

Let B be the denominator of the right-hand side of (3), then (2) and (3) imply that

$$|v_i| \leq 8B^3 n^{1/4} \quad i = 1, 2, 3, 4. \quad (4)$$

Remark 1. In our case, where $\Psi^2 + 1 = 0$ and $\Phi^2 + r\Phi + s = 0$, we get as norm function

$$\begin{aligned} & x_1^4 + s^2 x_2^4 + x_3^4 + s^2 x_4^4 - 2rx_1^3 x_2 - 2rsx_1 x_2^3 - 2rx_3^3 x_4 - 2rsx_3 x_4^3 + \\ & (r^2 + 2s)x_1^2 x_2^2 + 2x_1^2 x_3^2 + (r^2 - 2s)x_1^2 x_4^2 + (r^2 - 2s)x_2^2 x_3^2 + 2s^2 x_2^2 x_4^2 + (r^2 + 2s)x_3^2 x_4^2 \\ & - 2rx_1^2 x_3 x_4 - 2rsx_2^2 x_3 x_4 - 2rx_1 x_2 x_3^2 - 2rsx_1 x_2 x_4^2 + 8sx_1 x_2 x_3 x_4, \end{aligned}$$

and therefore

$$B = (4 + 4s^2 + 8s + 8|r| + 8|r|s + 2(r^2 + 2s) + 2|r^2 - 2s|)^{1/4}. \quad (5)$$

From (1) and (4) we have proved the following theorem.

Theorem 2. *Let E/\mathbb{F}_p be a GLV curve and E'/\mathbb{F}_{p^2} a twist, together with the two efficient endomorphisms Φ and Ψ , where everything is defined as at the start of Section 5. Suppose that the minimal polynomial of Φ is $X^2 + rX + s = 0$. Let $P \in E'(\mathbb{F}_{p^2})$ a generator of the large subgroup of prime order n . There exists an efficient algorithm, which for any $k \in [1, n]$ finds integers k_1, k_2, k_3, k_4 such that*

$$kP = k_1 P + k_2 \Phi(P) + k_3 \Psi(P) + k_4 \Psi\Phi(P) \quad \text{with } \max_i (|k_i|) \leq 16B^3 n^{1/4}$$

and

$$B = (4 + 4s^2 + 8s + 8|r| + 8|r|s + 2(r^2 + 2s) + 2|r^2 - 2s|)^{1/4}.$$

6 A Tale of Two Cornacchia Algorithms

In view of the fact that the LLL algorithm is rather inefficient compared to other dedicated algorithms in dimension less than five (running in $O(\log^3 n)$), we can ask ourselves if we can sharpen the bound of Theorem 2 and provide an explicit description of a simpler algorithm to find a short basis of $\ker F$. This is the scope of the the present section. Our algorithm has a running time of $O(\log^2 n)$, and will produce a proved bound greatly improving the $16B^3$ of Theorem 2.

The idea is to modify the original GLV approach which finds a short basis using an extended Euclidean algorithm. We find that in this case we need to perform two such algorithms, one in \mathbb{Z} , like in the GLV original paper, the other one in $\mathbb{Z}[i]$, the Gaussian integers. The main difficulty here lies in the correct choice of the remainders in the Gaussian gcd algorithm, since we don't have a canonical way to choose a "positive" one.

In contrast to Section 5, where we worked with generic endomorphisms Φ, Ψ generating a bi-quadratic field, we will strongly use here the fact that $\Psi^2 + 1 = 0$. We will denote indifferently by the letter i the usual imaginary root of unity in \mathbb{C} , the integer mod n such that $\Psi(P) = iP$, as well as the endomorphism Ψ . In particular, we let, for $z = a + ib \in \mathbb{Z}[i]$, $zP = aP + ibP = aP + b\Psi(P)$. The context in which we are referring to one or the other of these interpretations will be clear each time. These differences notwithstanding, we suppose that we are set as in the first paragraph of Section 5.

6.1 The Euclidean Algorithm in \mathbb{Z}

The first step is to find $\nu = a + ib \in \mathbb{Z}[i]$ such that $|\nu|^2 = a^2 + b^2 = n$, i.e. a Gaussian prime above n . Recall that n splits in $\mathbb{Z}[i]$. Let $\nu = a + ib$ a prime above n . We can furthermore assume that $\nu P = aP + biP = aP + b\Psi(P) = 0$, since $\nu\bar{\nu}P = nP = 0$ and hence either $\bar{\nu}P$ is a nonzero multiple of P and therefore $\nu P = 0$, or else we $\bar{\nu}P = 0$, so that in any case one of the Gaussian primes (WLOG ν) above n will have $\nu P = 0$. We can find ν by Cornacchia's algorithm [1, Section 1.5.2], which is a truncated form of the GLV algorithm. For completeness and consistency with what will follow, we recall how this is done.

Let $\mu \in [1, n]$ such that $\mu \equiv i \pmod{n}$, with i being defined by $\Psi(P) = iP$. Actually, in the GLS approach [2], it has been pointed out that this value of μ can be readily computed from $\#E(\mathbb{F}_p)$. The extended Euclidean algorithm to compute the gcd of n and μ produces three terminating sequences of integers $(r_j)_{j \geq 0}$, $(s_j)_{j \geq 0}$ and $(t_j)_{j \geq 0}$ such that

$$\begin{pmatrix} r_{j+2} & s_{j+2} & t_{j+2} \\ r_{j+1} & s_{j+1} & t_{j+1} \end{pmatrix} = \begin{pmatrix} -q_{j+1} & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} r_{j+1} & s_{j+1} & t_{j+1} \\ r_j & s_j & t_j \end{pmatrix}, \quad j \geq 0 \quad (6)$$

for some integer $q_{j+1} > 0$ and initial data

$$\begin{pmatrix} r_1 & s_1 & t_1 \\ r_0 & s_0 & t_0 \end{pmatrix} = \begin{pmatrix} \mu & 0 & 1 \\ n & 1 & 0 \end{pmatrix}. \quad (7)$$

This means that at step $j \geq 0$,

$$r_j = q_{j+1}r_{j+1} + r_{j+2}$$

and similarly for the other sequences. The sequence $(q_j)_{j \geq 1}$ is *uniquely* defined by imposing that the previous equation be the integer division of r_j by r_{j+1} . In other terms, $q_{j+1} = \lfloor r_j / r_{j+1} \rfloor$. This implies by induction that all the sequences are well defined in the integers, together with the following properties.

Lemma 2. *The sequences $(r_j)_{j \geq 0}$, $(s_j)_{j \geq 0}$ and $(t_j)_{j \geq 0}$ defined by (6) and (7) with $q_{j+1} = \lfloor r_j / r_{j+1} \rfloor$ satisfy the following properties, valid for all $j \geq 0$.*

1. $r_j > r_{j+1} \geq 0$ and $q_{j+1} \geq 1$,
2. $(-1)^j s_j \geq 0$ and $|s_j| < |s_{j+1}|$ (this last inequality valid for $j \geq 1$),
3. $(-1)^{j+1} t_j \geq 0$ and $|t_j| < |t_{j+1}|$,
4. $s_{j+1}r_j - s_j r_{j+1} = (-1)^{j+1} r_1$,
5. $t_{j+1}r_j - t_j r_{j+1} = (-1)^j r_0$,
6. $r_0 s_j + r_1 t_j = r_j$.

These properties lie at the heart of the original GLV algorithm. They imply in particular via 1. that the algorithm terminates (once r_j reaches zero), and that it has $O(\log n)$ steps, as $r_j = q_{j+1}r_{j+1} + r_{j+2} \geq r_{j+1} + r_{j+2} > 2r_{j+2}$. Note that 1., 2. & 3. imply that 4. & 5. can be rewritten in our case respectively as

$$|s_{j+1}r_j| + |s_j r_{j+1}| = \mu \quad \text{and} \quad |t_{j+1}r_j| + |t_j r_{j+1}| = n. \quad (8)$$

The Cornacchia (as well as the GLV) algorithm doesn't make use of the full sequences (r_j) , (s_j) and (t_j) but rather stops at the $m \geq 0$ such that $r_m \geq \sqrt{n}$ and $r_{m+1} < \sqrt{n}$. An application of (8) with $j = m$ yields $|t_{m+1}r_m| < n$ or $|t_{m+1}| < \sqrt{n}$. Since by 6. we have $r_{m+1} - \mu t_{m+1} = n s_{m+1} \equiv 0$

(mod n) we deduce that $r_{m+1}^2 + t_{m+1}^2 = (r_{m+1} - \mu t_{m+1})(r_{m+1} + \mu t_{m+1}) \equiv 0 \pmod{n}$. Moreover $t_{m+1} \neq 0$ by 3. so that $0 < r_{m+1}^2 + t_{m+1}^2 < n + n = 2n$ which therefore implies that $r_{m+1}^2 + t_{m+1}^2 = n$ and finally that $\nu = r_{m+1} - it_{m+1}$.

We present here the pseudo-code of this Euclidean algorithm in \mathbb{Z} .

Algorithm 1 (Cornacchia's GCD in \mathbb{Z})

Input: $n \equiv 1 \pmod{4}$ prime, $1 < \mu < n$ such that $\mu^2 \equiv -1 \pmod{n}$.

Output: $\nu = \nu_{(R)} + i\nu_{(I)}$ Gaussian prime dividing n , such that $\nu P = 0$.

1. initialize:

$r_0 \leftarrow n, r_1 \leftarrow \mu, r_2 \leftarrow n,$
 $t_0 \leftarrow 0, t_1 \leftarrow 1, t_2 \leftarrow 0,$
 $q \leftarrow 0.$

2. main loop:

while $r_2^2 \geq n$ **do**
 $q \leftarrow \lfloor r_0/r_1 \rfloor,$
 $r_2 \leftarrow r_0 - qr_1, r_0 \leftarrow r_1, r_1 \leftarrow r_2,$
 $t_2 \leftarrow t_0 - qt_1, t_0 \leftarrow t_1, t_1 \leftarrow t_2.$

3. return:

$\nu = r_1 - it_1, \nu_{(R)} = r_1, \nu_{(I)} = -t_1$

6.2 The Euclidean Algorithm in $\mathbb{Z}[i]$

In the previous subsection we have given a meaning to zP , where $z \in \mathbb{Z}[i]$, and we have seen how to construct ν , a Gaussian prime such that $\nu P = 0$. By identifying¹⁰ $(x_1, x_2, x_3, x_4) \in \mathbb{Z}^4$ with $(z_1, z_2) = (x_1 + ix_3, x_2 + ix_4) \in \mathbb{Z}[i]^2$, we can rewrite the 4-GLV reduction map F of Section 5 as (using the same letter F by abuse of notation)

$$F: \mathbb{Z}[i]^2 \rightarrow \mathbb{Z}[i]/\nu \cong \mathbb{Z}/n$$

$$(z_1, z_2) \mapsto z_1 + \lambda z_2 \pmod{\nu}.$$

This F should be confronted to the map \mathfrak{f} of Section 2. In mimicking the GLV original paper [3] we are drawn to applying the extended Euclidean algorithm (defined exactly as before, with integer divisions occurring in $\mathbb{Z}[i]$, henceforth denoted EGEA in short for extended Gaussian Euclidean algorithm) to the pair $(r_0, r_1) = (\lambda, \nu)$ if $\lambda \geq \sqrt{2}|\nu|$ and $(r_0, r_1) = (\lambda + n, \nu)$ otherwise (the latter case being exceptionally rare). We should note that 4., 5. & 6. of Lemma 2 still hold and 1. holds in modulus (in particular the algorithm terminates). However, in the analysis of this algorithm, especially in [9], a crucial rôle is played by (8), realising a bound on $|s_{j+1}r_j|$ and $|s_jr_{j+1}|$ out of a bound on

$$s_{j+1}r_j - s_jr_{j+1} = (-1)^{j+1}\nu \tag{9}$$

¹⁰ It is important to keep in mind that this association is only an isomorphism of abelian groups (\mathbb{Z} -modules). However, $\mathbb{Z}[i]^2$ is also endowed with a structure of $\mathbb{Z}[i]$ -module.

in the present case. This fact, as we saw, stems from the alternating sign of the sequence (s_j) , which results from taking a canonical form of integer division with positive quotients q_{j+1} and nonnegative remainders r_{j+2} , a property which is not available here. Nevertheless, we can still use a similar reasoning using (9), provided that the arguments of $s_{j+1}r_j$ and $s_j r_{j+1}$ are not too close, so as to avoid a high degree of cancellation.

The first observation is that in the case of Gaussian integers there can be 2, 3 or 4 possible choices for a remainder in the j -th step of the integer division $r_j = q_{j+1}r_{j+1} + r_{j+2}$. It turns out that choosing at each step $j \geq 0$ of the EGEA a remainder r_{j+2} with smallest modulus will yield the following decomposition theorem.

Theorem 3. *In the notations of Theorem 2, the lattice reduction consisting of Cornacchia's algorithm in \mathbb{Z} with positive remainders (Algorithm 1) and in $\mathbb{Z}[i]$ with smallest remainders (Algorithm 2 and 3) runs in $O(\log^2 n)$ binary operations and will result in a decomposition of any $k \in [1, n]$ into integers k_1, k_2, k_3, k_4 such that*

$$kP = k_1P + k_2\Phi(P) + k_3\Psi(P) + k_4\Psi\Phi(P)$$

with

$$\max_i (|k_i|) < 103 \left(\sqrt{1 + |r| + s} \right) n^{1/4}.$$

We give here the pseudo-code of Cornacchia's Algorithm in $\mathbb{Z}[i]$ in two forms, working with complex numbers and separating real and imaginary parts.

Algorithm 2 (Cornacchia's algorithm in $\mathbb{Z}[i]$ - compact form)

Input: ν Gaussian prime dividing n rational prime, $1 < \lambda < n$ such that $\lambda^2 + r\lambda + s \equiv 0 \pmod{n}$.

Output: Two $\mathbb{Z}[i]$ -linearly independent vectors v_1 & v_2 of $\ker F \subset \mathbb{Z}[i]^2$ of rectangle norms $< 51.5(\sqrt{1 + |r| + s}) n^{1/4}$.

1. initialize:

 If $\lambda^2 \geq 2n$ then

$r_0 \leftarrow \lambda,$

 else

$r_0 \leftarrow \lambda + n,$

$r_1 \leftarrow \nu, r_2 \leftarrow n,$

$s_0 \leftarrow 1, s_1 \leftarrow 0, s_2 \leftarrow 0,$

$q \leftarrow 0.$

2. main loop:

 while $|r_2|^4(1 + |r| + s)^2 \geq n$ do

$q \leftarrow$ closest Gaussian integer to $r_0/r_1,$

$r_2 \leftarrow r_0 - qr_1, r_0 \leftarrow r_1, r_1 \leftarrow r_2,$

$s_2 \leftarrow s_0 - qs_1, s_0 \leftarrow s_1, s_1 \leftarrow s_2.$

3. return:

$v_1 = (r_0, -s_0), v_2 = (r_1, -s_1)$

Algorithm 3 (Cornacchia's algorithm in $\mathbb{Z}[i]$ - real & imaginary parts)

Input: ν Gaussian prime dividing n rational prime, $1 < \lambda < n$ such that $\lambda^2 + r\lambda + s \equiv 0 \pmod{n}$.

Output: Four \mathbb{Z} -linearly independent vectors v_1, v_2, v_3 and $v_4 \in \ker F \subset \mathbb{Z}^4$ of rectangle norms $< 51.5(\sqrt{1+|r|+s})n^{1/4}$.

1. initialize:

If $\lambda^2 \geq 2n$ then

$r_{0,(R)} \leftarrow \lambda,$

else

$r_{0,(R)} \leftarrow \lambda + n,$

$r_{0,(I)} \leftarrow 0,$

$r_{1,(R)} \leftarrow \nu_{(R)}, r_{1,(I)} \leftarrow \nu_{(I)},$

$r_{2,(R)} \leftarrow n, r_{2,(I)} \leftarrow 0,$

$s_{0,(R)} \leftarrow 1, s_{0,(I)} \leftarrow 0,$

$s_{1,(R)} \leftarrow 0, s_{1,(I)} \leftarrow 0,$

$s_{2,(R)} \leftarrow 0, s_{2,(I)} \leftarrow 0,$

$q_{(R)} \leftarrow 0, q_{(I)} \leftarrow 0.$

2. main loop:

while $(r_{2,(R)}^4 + 2r_{2,(R)}^2 r_{2,(I)}^2 + r_{2,(I)}^4)(1 + |r| + s)^2 \geq n$ **do**

$$q_{(R)} \leftarrow \left\lfloor \frac{r_{0,(R)}r_{1,(R)} + r_{0,(I)}r_{1,(I)}}{r_{1,(R)}^2 + r_{1,(I)}^2} \right\rfloor,$$

$$q_{(I)} \leftarrow \left\lfloor \frac{r_{0,(I)}r_{1,(R)} - r_{0,(R)}r_{1,(I)}}{r_{1,(R)}^2 + r_{1,(I)}^2} \right\rfloor,$$

$r_{2,(R)} \leftarrow r_{0,(R)} - (q_{(R)}r_{1,(R)} - q_{(I)}r_{1,(I)}),$

$r_{2,(I)} \leftarrow r_{0,(I)} - (q_{(R)}r_{1,(I)} + q_{(I)}r_{1,(R)}),$

$r_{0,(R)} \leftarrow r_{1,(R)}, r_{1,(R)} \leftarrow r_{2,(R)},$

$r_{0,(I)} \leftarrow r_{1,(I)}, r_{1,(I)} \leftarrow r_{2,(I)},$

$s_{2,(R)} \leftarrow s_{0,(R)} - (q_{(R)}s_{1,(R)} - q_{(I)}s_{1,(I)}),$

$s_{2,(I)} \leftarrow s_{0,(I)} - (q_{(R)}s_{1,(I)} + q_{(I)}s_{1,(R)}),$

$s_{0,(R)} \leftarrow s_{1,(R)}, s_{1,(R)} \leftarrow s_{2,(R)},$

$s_{0,(I)} \leftarrow s_{1,(I)}, s_{1,(I)} \leftarrow s_{2,(I)}.$

3. return:

$v_1 = (r_{0,(R)}, -s_{0,(R)}, r_{0,(I)}, -s_{0,(I)}), v_2 = (r_{1,(R)}, -s_{1,(R)}, r_{1,(I)}, -s_{1,(I)}),$

$v_3 = (-r_{0,(I)}, s_{0,(I)}, r_{0,(R)}, -s_{0,(R)}), v_4 = (-r_{1,(I)}, s_{1,(I)}, r_{1,(R)}, -s_{1,(R)}).$

7 Proof of Theorem 3

This section is devoted to proving that Algorithms 2 and 3 produce a reduced basis of $\ker F$ of rectangle norm $< 51.5(\sqrt{1+|r|+s})n^{1/4}$. The proof of the decomposition of k follows from the deduction recalled in Section 5.

Let us note first, about the running time, that it is known that the extended Euclidean algorithm runs in $O(\log^2 n)$ bits. The same analysis will also show that its Gaussian version runs in $O(\log^2 n)$ bits, since its number of steps is also logarithmic. In short, this works as follows: if $b_j = \lfloor \log_2(|r_j|) \rfloor$ (i.e. the bitsize of $|r_j|$), then step j of the EGEA necessitates to find q_{j+1} and then r_{j+2} . One can show that integer division of two h -bit Gaussian integers with a ℓ -bit quotient runs in $O(h(\ell + 1))$ binary operations. Finding q_{j+1} has therefore a runtime $O(b_j(c_{j+1} + 1))$, where $c_{j+1} = \lfloor \log_2(|q_{j+1}|) \rfloor = b_j - b_{j+1} + O(1)$. Similarly, knowing q_{j+1} , computing r_{j+2} can be done in $O(b_{j+1}c_{j+1}) + O(b_{j+1}) = O(b_{j+1}(b_j - b_{j+1})) + O(b_{j+1})$. If $S = O(\log n)$ is the number of steps of the EGEA, the total runtime is less than a constant times

$$\sum_{j=0}^S b_j(b_j - b_{j+1}) + b_j = O(b_0^2 + b_0 S) = O(\log^2 n) .$$

In the following, whenever $z \in \mathbb{C}^*$, its argument value $\arg(z)$ will be always chosen in $(-\pi, \pi]$. By *lattice square* we mean a square of side length one with vertices in $\mathbb{Z}[i]$. We single out eight *exceptional lattice squares*, which are those lattice squares with a vertex of modulus 1 (that is ± 1 or $\pm i$) but not containing the origin as a vertex. Our analysis of the EGEA rests on the following lemmas.

Lemma 3 (A geometric property of squares). *There exists an absolute real constant $\theta \approx 2.45861$ (with $2 \arctan 2 < \theta$) such that, for any point P of a lattice square, different from the vertices, letting V_1 be the closest vertex to P , there exists another vertex $V_2 \neq V_1$ with $\theta \leq \widehat{V_1 P V_2} \leq \pi$. (Note that $V_1 P \leq 1/\sqrt{2}$.)*

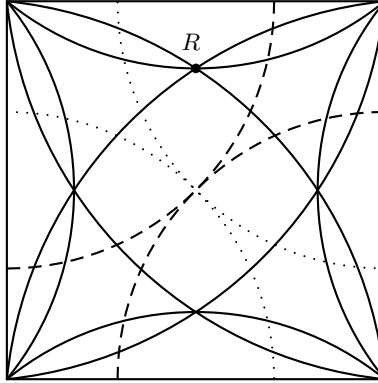


Fig. 1.

Proof. This is one case where a picture is worth one thousand words. We refer to Figure 1 for a visual explanation of why the argument works. The dotted and dashed circle arcs are centred on the vertices and have radius $1/\sqrt{2}$. The plain circle arcs have the following property: for any point P on them, the two square vertices V and V' belonging to them make an angle of θ with P , in other terms $\widehat{V P V'} = \theta$. Therefore points between two bigger arcs (in one of the two almond-shaped

regions) “look” at the diagonally opposite vertices marking the intersection of these arcs with an angle between θ and π . We then choose the closest vertex to get a distance $\leq 1/\sqrt{2}$. In case P is at the intersection of the two almond-shaped regions (in the “blown square”), we may have to choose one region where one of the vertices is at distance $\leq 1/\sqrt{2}$, but this is always possible, since the dashed and dotted disks cover everything. Finally, if P does not belong to the union of the two almond-shaped regions, then it lies inside one of the smaller plain disks, where its angle between two appropriate consecutive vertices will also be between θ and π . Furthermore, by choosing the closest vertex V_1 to P , we have $V_1P < 1/\sqrt{2}$. \square

It remains to explain how we can calculate θ , or rather its value on the usual trigonometric functions $\sin \theta$ and $\cos \theta$ (which is what we really need later), since we can show that they are algebraic numbers expressible by radicals, but $\theta/\pi \notin \mathbb{Q}$.

We concentrate on finding the cartesian coordinates of $R = (1/2, 1-u/2)$, appearing in Figure 1, supposing the vertices are the origin, $(1,0)$, $(1,1)$ and $(0,1)$. Our aim is then to find $u \in (0,1)$. A look at Figure 2 shows the disposition of the angles, so that $u = \cot(\theta/2)$ and $2-u = \cot(3\theta/2-\pi) = \cot(3\theta/2)$. The triplication formulas for the cotangent then show that u satisfies the equation

$$u + \frac{3u - u^3}{1 - 3u^2} = 2 \iff 2u^3 - 3u^2 - 2u + 1 = 0 .$$

Solving it yields that the root we are looking for is

$$u = \frac{(\sqrt{3}i + 1)(12\sqrt{237}i - 54)^{\frac{1}{3}}}{122^{\frac{1}{3}}} + \frac{72^{\frac{1}{3}}(1 - \sqrt{3}i)}{4(12\sqrt{237}i - 54)^{\frac{1}{3}}} + \frac{1}{2} \approx 0.3554157$$

where the determination of the cube root is the one in the first quadrant.

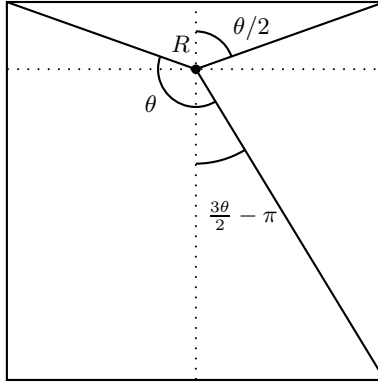


Fig. 2.

Remark 2. One can see that $\theta/\pi \notin \mathbb{Q}$ in the following way.

$$\cot(\theta/2) = i \frac{e^{i\theta/2} + e^{-i\theta/2}}{e^{i\theta/2} - e^{-i\theta/2}}$$

and supposing by absurd that $\theta/\pi \in \mathbb{Q}$ we would have that $e^{i\theta}$ is a root of unity. The preceding equality shows that then $\cot(\theta/2)$ belongs to a cyclotomic extension of \mathbb{Q} , whose Galois group is abelian. But we have seen that the irreducible polynomial of $\cot(\theta/2)$ is $2x^3 - 3x^2 - 2x + 1$, with discriminant 316, not a rational square. Therefore its Galois group is the nonabelian S_3 , contradiction.

Remark 3. When applying Lemma 3, it is essential that we be able to choose from the set of all vertices of the lattice square which ones are the adequate V_1 and V_2 . Since the only excluded quotient q_j is zero, it means that we must be careful to avoid all four squares which have the origin as a vertex. But this follows from the fact that at all steps $j \geq 0$ we always have $|r_j/r_{j+1}| \geq \sqrt{2}$.

Define $\Theta = \arctan 2 - \pi/3$ and $A = 1/\sin \Theta = 2\sqrt{5}(8 + 5\sqrt{3})/\sqrt{13 + 4\sqrt{3}} \approx 16.6902$. In the following analysis of the EGEA, it will be useful to make the following distinction between indices.

Definition 1 (Good and bad j 's). A step $j \geq 0$ of the EGEA will be called *bad* if, during the $j - 1$ -th step, among all four choices of q_j as a vertex of the lattice square containing r_{j-1}/r_j (and consequent choice of r_{j+1} and s_{j+1} , noting that for the purpose of this definition we do not require that $|r_{j+1}| < |r_j|$), we always have $s_j s_{j+1} r_{j+1} \neq 0$ and

$$\left| \arg \left(\frac{s_{j+1} r_j}{s_j r_{j+1}} \right) \right| < \Theta .$$

Otherwise j is called *good*.

Remark 4. Note that $j = 0$ and $j = 1$ are always good, since $s_1 = 0$.

Lemma 4 (Use of good j 's). If j is good then for some choice of r'_{j+1} (and relative s'_{j+1}) we have

$$|s'_{j+1} r_j - s_j r'_{j+1}| \geq \sin \Theta \max(|s'_{j+1} r_j|, |s_j r'_{j+1}|)$$

and, therefore, if we choose a r_{j+1} with smallest modulus, then

$$\max(|s_j r_{j+1}|, |s_{j+1} r_j|) \leq (A + 1)|\nu|$$

Proof. Notice that the result holds trivially if $s_j s'_{j+1} r'_{j+1} = 0$. Otherwise, this is a straightforward application of a general inequality about complex numbers that we can express as follows: let $\zeta \in \mathbb{C}^*$ with $\pi \geq |\arg(\zeta)| \geq \Theta$. We claim that under these conditions, $|1 - \zeta| \geq \sin \Theta$. Indeed, writing $\zeta = r e^{i\psi}$ with $\Theta \leq \psi \leq \pi$ we have

$$|1 - \zeta|^2 = (1 - r e^{i\psi})(1 - r e^{-i\psi}) = 1 - 2r \cos \psi + r^2 .$$

First note that we can suppose $\psi \leq \pi/2$, otherwise clearly $|1 - \zeta| \geq 1$. The last expression in r , when viewed as a quadratic polynomial has minimum (over \mathbb{R}) equal to $-\Delta/4 = -(4 \cos^2 \psi - 4)/4 = \sin^2 \psi \geq \sin^2 \Theta$. Therefore $|1 - \zeta| \geq \sin \Theta$ thereby proving our claim. The first part of the lemma will follow by applying the claim to $\zeta = s'_{j+1} r_j / s_j r'_{j+1}$ and $\zeta = s_j r'_{j+1} / s'_{j+1} r_j$ successively.

The second part follows from

$$|s_j r_{j+1}| \leq |s_j r'_{j+1}| \leq A |s'_{j+1} r_j - s_j r'_{j+1}| = A |\nu|$$

and therefore

$$|s_{j+1} r_j| = |s_{j+1} r_j - s_j r_{j+1} + s_j r_{j+1}| \leq |s_{j+1} r_j - s_j r_{j+1}| + |s_j r_{j+1}| \leq |\nu| + A |\nu|$$

□

Remark 5. We have seen in the course of the proof the preceding lemma the following fact: if $\zeta \in \mathbb{C}^*$ with $\pi \geq |\arg(\zeta)| \geq \psi$, then $|1 - \zeta| \geq \sin \psi$. This is equivalent to the following assertion (set $\zeta = 1 - \xi$), used in the proof of the next lemma: if $|\xi| < \sin \psi$, then $|\arg(1 - \xi)| < \psi$.

The next result is crucial in controlling what happens when things go “uncontrolled”. Its proof is rather elaborate.

Lemma 5 (Bad- j behaviour of s_j). *If j is bad, then*

$$|s_{j+1}| \leq 2\sqrt{2}|s_{j-1}| \quad \text{and} \quad |s_j| \leq |s_{j-1}| \quad .$$

Proof. We first suppose that the point P of affix r_{j-1}/r_j does not belong to an exceptional lattice square. Let V_1 and V_2 as in Lemma 3 of affixes respectively q_j and q'_j . Upon defining $r'_{j+1} = r_{j-1} - q'_j r_j$, since $r_{j+1} = r_{j-1} - q_j r_j$, Lemma 3 states that $\pi \geq |\arg((q_j - r_{j-1}/r_j)/(q'_j - r_{j-1}/r_j))| = |\arg(r_{j+1}/r'_{j+1})| \geq \theta$. By definition of “bad” we have, denoting $s'_{j+1} = s_{j-1} - q'_j s_j$,

$$\left| \arg \left(\frac{s_{j+1} r_j}{s_j r_{j+1}} \right) \right| < \Theta \quad \text{and} \quad \left| \arg \left(\frac{s'_{j+1} r_j}{s_j r'_{j+1}} \right) \right| < \Theta \quad ,$$

and this yields

$$\begin{aligned} \left| \arg \left(\frac{s_{j+1} r'_{j+1}}{s'_{j+1} r_{j+1}} \right) \right| &= \left| \arg \left(\frac{s_{j+1} r_j}{s_j r_{j+1}} \right) + \arg \left(\frac{s_j r'_{j+1}}{s'_{j+1} r_j} \right) \right| \\ &\leq \left| \arg \left(\frac{s_{j+1} r_j}{s_j r_{j+1}} \right) \right| + \left| \arg \left(\frac{s_j r'_{j+1}}{s'_{j+1} r_j} \right) \right| < 2\Theta \quad . \end{aligned}$$

We deduce

$$\begin{aligned} \left| \arg \left(\frac{s_{j+1} r'_{j+1}}{s'_{j+1} r_{j+1}} \right) + \arg \left(\frac{r_{j+1}}{r'_{j+1}} \right) \right| &\geq \left| \left| \arg \left(\frac{r_{j+1}}{r'_{j+1}} \right) \right| - \left| \arg \left(\frac{s_{j+1} r'_{j+1}}{s'_{j+1} r_{j+1}} \right) \right| \right| \\ &> \theta - 2\Theta > \frac{2\pi}{3} \quad , \end{aligned}$$

while on the other hand

$$\left| \arg \left(\frac{s_{j+1} r'_{j+1}}{s'_{j+1} r_{j+1}} \right) \right| + \left| \arg \left(\frac{r_{j+1}}{r'_{j+1}} \right) \right| < 2\Theta + \pi < \theta - \frac{2\pi}{3} + \pi < \frac{4\pi}{3}$$

which together imply

$$\left| \arg \left(\frac{s_{j+1}}{s'_{j+1}} \right) \right| > \frac{2\pi}{3} \quad . \quad (10)$$

Now assume that $|s_j| > |s_{j-1}|$. Then $|q_j s_j| > \sqrt{2}|s_{j-1}|$ and $|q'_j s_j| > \sqrt{2}|s_{j-1}|$, since the quotients q_j, q'_j Gaussian integers of modulus different from zero or one. Furthermore, since there is at most one Gaussian integer of modulus one in a lattice square, we have that either $|q_j s_j| > 2|s_{j-1}|$ or $|q'_j s_j| > 2|s_{j-1}|$. Therefore, by Remark 5,

$$\left| \arg \left(\frac{q_j s_j - s_{j-1}}{q_j s_j} \right) \right| = \left| \arg \left(1 - \frac{s_{j-1}}{q_j s_j} \right) \right| \leq \frac{\pi}{4}$$

and similarly

$$\left| \arg \left(\frac{q'_j s_j - s_{j-1}}{q'_j s_j} \right) \right| \leq \frac{\pi}{4} ,$$

with at least one of them being $\leq \pi/6$. We then get, using that $|\arg(q_j/q'_j)| \leq \pi/4$,

$$\begin{aligned} \left| \arg \left(\frac{s_{j+1}}{s'_{j+1}} \right) \right| &= \left| \arg \left(\frac{q_j s_j - s_{j-1}}{q_j s_j} \right) + \arg \left(\frac{q_j}{q'_j} \right) + \arg \left(\frac{q'_j s_j}{q'_j s_j - s_{j-1}} \right) \right| \\ &\leq \frac{\pi}{4} + \frac{\pi}{4} + \frac{\pi}{6} = \frac{2\pi}{3} \end{aligned}$$

contradicting (10).

Exceptional Choices of V_1 and V_2 :

We now discuss the exceptional cases when q_j or $q'_j = \pm 1, \pm i$, which need to be handled *ad hoc*. By symmetry (without loss of generality), we place ourselves in the case when r_{j-1}/r_j lies in the lattice square of vertices $i, 1+i, 1+2i, 2i$. It then belongs to one of the five regions labeled 1 to 5 on Figure 3. Note that the open grey region is off-limits, since $|r_{j-1}/r_j| \geq \sqrt{2}$. Each of these regions contains two lattice points, which as before we will denote V_1 for the one closest to the point P of affix r_{j-1}/r_j and V_2 for the other one (in case P lies on the boundary between two or more zones, their distinction is immaterial). Note that V_1 is closest to P among *all* four vertices.

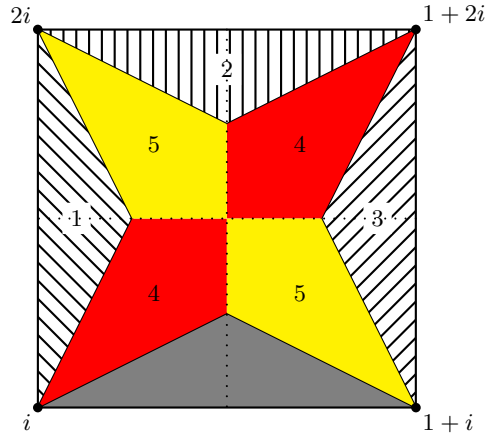


Fig. 3.

Region 1 (delimited by a triangle of vertices $i, 1/4 + 3i/2, 2i$): In this case, $|\arg(r_{j+1}/r'_{j+1})| = \widehat{V_1 P V_2} \geq 2 \arctan 2$. Supposing to fix notations that $q_j = i$ and $q'_j = 2i$ we have, assuming that $|s_j| > |s_{j-1}|$ and using Remark 5

$$\left| \arg \left(\frac{q_j s_j - s_{j-1}}{q_j s_j} \right) \right| \leq \frac{\pi}{2} , \quad \left| \arg \left(\frac{q'_j s_j - s_{j-1}}{q'_j s_j} \right) \right| \leq \frac{\pi}{6} .$$

On the other hand, a reasoning similar to the one leading to (10) with the value $2 \arctan 2$ instead of θ will show that again $|\arg(s_{j+1}/s'_{j+1})| > 2\pi/3$, which leads to a contradiction since

$$\begin{aligned} \left| \arg \left(\frac{s_{j+1}}{s'_{j+1}} \right) \right| &= \left| \arg \left(\frac{q_j s_j - s_{j-1}}{q_j s_j} \right) + \arg \left(\frac{q_j}{q'_j} \right) + \arg \left(\frac{q'_j s_j}{q'_j s_j - s_{j-1}} \right) \right| \\ &\leq \frac{\pi}{2} + 0 + \frac{\pi}{6} = \frac{2\pi}{3} . \end{aligned}$$

The other four cases are treated similarly and we briefly outline them.

Region 2 (delimited by a triangle of vertices $2i, 1+7i/4, 1+2i$): Here $|\arg(r_{j+1}/r'_{j+1})| \geq 2 \arctan 2$. Letting $q_j = 2i, q'_j = 1+2i$ one can show, assuming that $|s_j| > |s_{j-1}|$, that

$$\begin{aligned} \frac{2\pi}{3} &< \left| \arg \left(\frac{s_{j+1}}{s'_{j+1}} \right) \right| = \left| \arg \left(\frac{q_j s_j - s_{j-1}}{q_j s_j} \right) + \arg \left(\frac{q_j}{q'_j} \right) + \arg \left(\frac{q'_j s_j}{q'_j s_j - s_{j-1}} \right) \right| \\ &\leq \frac{\pi}{6} + \left(\frac{\pi}{2} - \arctan 2 \right) + \left(\frac{\pi}{2} - \arctan 2 \right) < \frac{\pi}{2} < \frac{2\pi}{3} , \end{aligned}$$

contradiction.

Region 3 (delimited by a triangle of vertices $1+2i, 3/4+i/2, 1+i$): Here $|\arg(r_{j+1}/r'_{j+1})| \geq 2 \arctan 2$. Letting $q_j = 1+i, q'_j = 1+2i$ one can show, assuming that $|s_j| > |s_{j-1}|$, that

$$\begin{aligned} \frac{2\pi}{3} &< \left| \arg \left(\frac{s_{j+1}}{s'_{j+1}} \right) \right| = \left| \arg \left(\frac{q_j s_j - s_{j-1}}{q_j s_j} \right) + \arg \left(\frac{q_j}{q'_j} \right) + \arg \left(\frac{q'_j s_j}{q'_j s_j - s_{j-1}} \right) \right| \\ &\leq \frac{\pi}{4} + \arctan(1/3) + \left(\frac{\pi}{2} - \arctan 2 \right) = \frac{\pi}{2} < \frac{2\pi}{3} , \end{aligned}$$

contradiction.

Region 4 (the red zone): Here $|\arg(r_{j+1}/r'_{j+1})| \geq \pi - \arctan 2 + \arctan(2/3)$. Letting $q_j = i, q'_j = 1+2i$ one can show, assuming that $|s_j| > |s_{j-1}|$, that

$$\begin{aligned} \frac{5\pi}{3} - 3 \arctan(2) + \arctan(2/3) &< \left| \arg \left(\frac{s_{j+1}}{s'_{j+1}} \right) \right| \\ &= \left| \arg \left(\frac{q_j s_j - s_{j-1}}{q_j s_j} \right) + \arg \left(\frac{q_j}{q'_j} \right) + \arg \left(\frac{q'_j s_j}{q'_j s_j - s_{j-1}} \right) \right| \\ &\leq \frac{\pi}{2} + \left(\frac{\pi}{2} - \arctan 2 \right) + \left(\frac{\pi}{2} - \arctan 2 \right) \\ &< \frac{5\pi}{3} - 3 \arctan(2) + \arctan(2/3) , \end{aligned}$$

contradiction.

Region 5 (the yellow zone): Here $|\arg(r_{j+1}/r'_{j+1})| \geq \pi - \arctan(4/7)$. Letting $q_j = 1 + i$, $q'_j = 2i$ one can show, assuming that $|s_j| > |s_{j-1}|$, that

$$\begin{aligned} \frac{5\pi}{3} - 3\arctan(2) + \arctan(2/3) &< \left| \arg \left(\frac{s_{j+1}}{s'_{j+1}} \right) \right| \\ &= \left| \arg \left(\frac{q_j s_j - s_{j-1}}{q_j s_j} \right) + \arg \left(\frac{q_j}{q'_j} \right) + \arg \left(\frac{q'_j s_j}{q'_j s_j - s_{j-1}} \right) \right| \\ &\leq \frac{\pi}{4} + \frac{\pi}{4} + \frac{\pi}{6} = \frac{2\pi}{3} \\ &< \frac{5\pi}{3} - 3\arctan(2) + \arctan(2/3) , \end{aligned}$$

contradiction.

We have thus proved that in any case $|s_j| \leq |s_{j-1}|$. To show the first part of the lemma, we proceed similarly, although there is a slight difference. We assume at first that $|s_{j+1}| > 2|s_{j-1}|$ and $|s'_{j+1}| > \sqrt{2}|s_{j-1}|$. Then, by Remark 5,

$$\left| \arg \left(\frac{s_{j+1} - s_{j-1}}{s_{j+1}} \right) \right| = \left| \arg \left(1 - \frac{s_{j-1}}{s_{j+1}} \right) \right| \leq \frac{\pi}{6}$$

and

$$\left| \arg \left(\frac{s'_{j+1} - s_{j-1}}{s'_{j+1}} \right) \right| \leq \frac{\pi}{4} .$$

Proceeding as previously,

$$\begin{aligned} \left| \arg \left(\frac{s_{j+1}}{s'_{j+1}} \right) \right| &= \left| \arg \left(\frac{s_{j+1}}{s_{j+1} - s_{j-1}} \right) + \arg \left(\frac{q_j}{q'_j} \right) + \arg \left(\frac{s'_{j+1} - s_{j-1}}{s'_{j+1}} \right) \right| \\ &\leq \frac{\pi}{4} + \frac{\pi}{4} + \frac{\pi}{6} = \frac{2\pi}{3} \end{aligned}$$

again contradicting (10), which also holds in the exceptional cases, as we have just seen. Therefore $|s_{j+1}| \leq 2|s_{j-1}|$ or $|s'_{j+1}| \leq \sqrt{2}|s_{j-1}|$ (or both). In the first case, we are done. Otherwise, since $s_{j+1} + q_j s_j = s'_{j+1} + q'_j s_j = s_{j-1}$, we derive

$$|s_{j+1}| \leq |s'_{j+1}| + |q_j - q'_j| |s_j| \leq \sqrt{2}|s_{j-1}| + \sqrt{2}|s_{j-1}| = 2\sqrt{2}|s_{j-1}| ,$$

by the already proved second part of the lemma and the fact that q_j, q'_j correspond to two vertices of the same lattice square, so that $|q_j - q'_j| \leq \sqrt{2}$. \square

Lemma 6 (Lower bound on generic vectors of $\ker F$). *For any nonzero $(z_1, z_2) \in \ker F$ we have*

$$\max(|z_1|, |z_2|) \geq \frac{\sqrt{|\nu|}}{\sqrt{1 + |r| + s}} .$$

In particular, for any $j \geq 0$ we have

$$\max(|r_j|, |s_j|) \geq \frac{\sqrt{|\nu|}}{\sqrt{1 + |r| + s}} .$$

Proof. This proof uses an argument already appearing in the proof of the original GLV algorithm, see [9], as well as Lemma 1. If $(0, 0) \neq (z_1, z_2) \in \ker F$ then $z_1 + \lambda z_2 \equiv 0 \pmod{\nu}$. If λ' is the other root of $X^2 + rX + s \pmod{n}$, we get that

$$z_1^2 - rz_1z_2 + sz_2^2 \equiv (z_1 + \lambda z_2)(z_1 + \lambda' z_2) \equiv 0 \pmod{\nu}.$$

Since $X^2 + rX + s$ is irreducible in $\mathbb{Q}(i)$ because the two quadratic fields are linearly disjoint, we therefore have $|z_1^2 - rz_1z_2 + sz_2^2| \geq |\nu|$. On the other hand if

$$\max(|z_1|, |z_2|) < \frac{\sqrt{|\nu|}}{\sqrt{1 + |r| + s}},$$

then

$$|z_1^2 - rz_1z_2 + sz_2^2| \leq |z_1|^2 + |r||z_1||z_2| + s|z_2|^2 < |\nu|,$$

a contradiction. To show the second part, it suffices to note that since $r_0s_j + \nu t_j = r_j$ (where, as mentioned previously, $r_0 = \lambda$ or $\lambda + n$), we have that

$$0 \equiv \nu t_j = r_j - r_0s_j \equiv r_j - \lambda s_j \pmod{\nu}$$

so that $(r_j, -s_j) \in \ker F$ for every $j \geq 0$. □

Proof (of Theorem 3). It remains here to show the improved bound, which brings us to finding four \mathbb{Q} -linearly independent vectors of $\ker F$ of rectangle norm bounded by $Cn^{1/4}$. Define $m \geq 1$ as the index such that

$$|r_m| \geq \frac{\sqrt{|\nu|}}{\sqrt{1 + |r| + s}} \quad \text{and} \quad |r_{m+1}| < \frac{\sqrt{|\nu|}}{\sqrt{1 + |r| + s}}. \quad (11)$$

Let us consider an index $j \leq m$. If it's good, then by Lemma 4 we have $|s_{j+1}r_j| \leq (A + 1)|\nu|$ and therefore, since $(|r_j|)$ is a decreasing sequence,

$$|s_{j+1}| \leq 2\sqrt{2}(A + 1)\sqrt{1 + |r| + s}\sqrt{|\nu|}. \quad (12)$$

On the other hand if it's bad, then let $l < j$ be the largest good index less than j . By Lemma 5 and Lemma 4 we have

$$\begin{aligned} \frac{|s_{j+1}|}{2\sqrt{2}} &\leq |s_{j-1}| \leq |s_{j-2}| \leq \dots \leq |s_l| \leq (A + 1) \frac{|\nu|}{|r_{l+1}|} \\ &\leq (A + 1)\sqrt{1 + |r| + s}\sqrt{|\nu|}, \end{aligned}$$

therefore in any case (12) holds. Applying this to $j = m - 1$ and $j = m$ we find that

$$\max(|s_m|, |s_{m+1}|) \leq 2\sqrt{2}(A + 1)\sqrt{1 + |r| + s}\sqrt{|\nu|}. \quad (13)$$

Moreover, using

$$s_{m+1}r_m - s_mr_{m+1} = (-1)^{m+1}\nu \quad (14)$$

and from (11), (13) we deduce

$$|s_{m+1}r_m| \leq |\nu| + |s_mr_{m+1}| \leq |\nu| + 2\sqrt{2}(A + 1)|\nu|.$$

In addition, by Lemma 6 we must have

$$|s_{m+1}| \geq \frac{\sqrt{|\nu|}}{\sqrt{1+|r|+s}}$$

which therefore implies that

$$|r_m| \leq (2\sqrt{2}(A+1)+1)\sqrt{1+|r|+s}\sqrt{|\nu|}.$$

This last equation, together with (13) and (11), show that the two vectors $v_1 = (r_m, -s_m), v_2 = (r_{m+1}, -s_{m+1}) \in \ker F \subset \mathbb{Z}[i]^2$ have rectangle norms bounded by $C\sqrt{|\nu|} = Cn^{1/4}$, for $C = (2\sqrt{2}(A+1)+1)\sqrt{1+|r|+s} < 51.5\sqrt{1+|r|+s}$ (that these two vectors belong to $\ker F$ was shown in the proof of Lemma 6).

We can find two more vectors by noticing that (14) implies that v_1 and v_2 are $\mathbb{Q}(i)$ -linearly independent. Therefore, the vectors $v_1, v_2, v_3 = iv_1, v_4 = iv_2$ are \mathbb{Q} -linearly independent. They all belong to $\ker F$ and have rectangle norms bounded by $Cn^{1/4}$. In view of the fact that the Euclidean norm upper-bounds the rectangle norm, the corresponding vectors in \mathbb{Z}^4 also have rectangle norms bounded by $Cn^{1/4}$, thus concluding the proof of the theorem, since these are exactly the four vectors returned by Algorithm 3. \square

Remark 6. Let us note that since we are in dimension less than 5, Nguyễn and Stehlé [11] have also produced an algorithm which finds vectors of successive minima in $\ker F$ with a running time $O(\log^2 n)$. However it doesn't seem to give an explicit bound on their length applicable to our case.

Remark 7. One may doubt about the pertinence of securing a faster lattice reduction algorithm for the GLV lattice $\ker F$. Indeed, at the present moment, the GLV method has been applied by choosing a fixed curve in the parameters and performing the lattice reduction offline. However, it is quite possible that in the future some new cryptosystem will require an online curve agreement, by counting points over a suitable field and successively performing the lattice reduction. For this, and human tendency of pushing away limitations, we consider that our previous argument, in addition to its formal elegance, may eventually find a useful application.

8 Performance Estimates

In this section, we assess performance of the four-dimensional GLV method in comparison with the traditional and two-dimensional cases. For our analysis, let us consider the curve $E : y^2 = x^3 + b$ over a quadratic extension field of large prime characteristic, exploiting a pseudo-Mersenne prime p such that -1 is a quadratic non-residue mod p , for efficiency purposes. Let us define the following notation: i) M, S, A and I represent field multiplication, squaring, addition and inversion over \mathbb{F}_p , respectively, and ii) m, s, a and i represent the same operations over \mathbb{F}_{p^2} . On the curve above, the expected costs of scalar multiplication at the 128-bit security level in terms of \mathbb{F}_{p^2} operations on one processor core are given by

- One core non-GLV: $256\text{DBL} + 42.5\text{mADD} + C_{\text{Precomp}} + C_{\text{Affine}} = 1108m + 1152s + 2090a + (1i + 64m + 19s + 56a) + (1i + 3m + 1s) = 2i + 1175m + 1172s + 2146a,$
- One core 2-GLV: $128\text{DBL} + 43.5\text{mADD} + C_{\text{Precomp}} + C_{\text{Affine}} = 732m + 643s + 1201a + (1i + 78m + 19s + 63a) + (1i + 3m + 1s) = 2i + 813m + 663s + 1264a,$

- One core 4-GLV: $64\text{DBL} + 45.5\text{mADD} + C_{\text{Precomp}} + C_{\text{Affine}} = 556m + 393s + 767a + (1i + 106m + 19s + 77a) + (1i + 3m + 1s) = 2i + 665m + 413s + 844a$,

corresponding to the fully sequential executions without using GLV, using 2-GLV and using 4-GLV, respectively. Note that DBL and mADD represent point doubling and mixed addition, and that their costs are $3m + 4s + 7a$ and $8m + 3s + 7a$ when using Jacobian coordinates. C_{Precomp} and C_{Affine} represent the cost of precomputation and final conversion to affine coordinates, respectively. The costs above assume the use of interleaving (INT) [3] with width- w non-adjacent form ($w\text{NAF}$) using 7 precomputed points ($w = 5$) and the use of the LM scheme for precomputing those points [8].

Similarly, the expected cost of 4-GLV on four cores is

- Four core 4-GLV: $64\text{DBL} + 12.5\text{mADD} + C_{\text{Precomp}} + C_{\text{Affine}} = 292m + 294s + 536a + (1i + 69m + 19s + 58a) + (1i + 3m + 1s) = 2i + 364m + 314s + 594a$.

Thus, it can be seen a steady cost reduction when switching from a non-GLV based implementation to 2- and 4-GLV. A similar improvement is observed when increasing the number of cores in the case of 4-GLV. Optimal performance is ultimately achieved with this method running on four cores. For instance, following implementation results on an AMD Phenom II X4 940, we have that $1i \approx 50m$, $1s \approx 0.70m$ and $1a \approx 0.2m$ when using a 127-bit prime. In this case, non-GLV, 2-GLV and 4-GLV cost $2525m$, $1630m$ and $1223m$ on one core, respectively. On the other hand, four core 4-GLV costs $803m$. Hence, a remarkable 3x speedup is expected when using 4-GLV on four cores in comparison with a traditional execution on one core.

Note that, in comparison to \mathbb{F}_p^2 arithmetic using a 128-bit prime (such as the one used in [4]), \mathbb{F}_p^2 multiplication is expected to be faster since internal field additions can be carried out without carry checks and lazy reduction applies efficiently (see also [7]); however, other operations get slightly more expensive because reduction involves a few extra shift and rotate operations.

Let us compare performance against a similar curve using the original GLV method over \mathbb{F}_p . In this case the expected costs when using one and two cores are ($1I \approx 200M$, $1S \approx 0.85M$ and $1A \approx 0.2M$ on the targeted platform)

- One core standard 2-GLV: $128\text{DBL} + 37.2\text{ADD} + 5.3\text{mADD} + C_{\text{Precomp}} + C_{\text{Affine}} = 836M + 640S + 1194A + (51M + 26S + 56A) + (1I + 3M + 1S) = 1I + 890M + 667S + 1250A = 1907M$,
- Two core standard 2-GLV: $128\text{DBL} + 19\text{ADD} + 3.5\text{mADD} + C_{\text{Precomp}} + C_{\text{Affine}} = 621M + 580S + 1054A + (44M + 26S + 56A) + (1I + 3M + 1S) = 1I + 668M + 607S + 1110A = 1606M$,

where DBL, mADD and ADD cost $3M + 4S + 7A$, $8M + 3S + 7A$ and $11M + 3S + 7A$ for the case of Jacobian coordinates. We assume the use of $w\text{NAF}$ with window width $w = 5$ and the LM precomputation scheme without inversions [6, Ch. 3]. Since in our case we have observed that in practice $1M \approx 0.75m$, the scaled costs of one core and two core standard 2-GLV are equivalent to $1430m$ and $1205m$, respectively. This means that on one core the 4-GLV method is expected to compute scalar multiplication in about 0.86 the time of the standard 2-GLV. Similarly, an optimal execution of 4-GLV on four cores is expected to run in about 0.67 the time of the optimal execution of the standard 2-GLV on two cores.

To confirm our findings we implemented the proposed method using the quadratic twist of E_1 over $\mathbb{F}_{p_1^2}$ given by $E'_1/\mathbb{F}_{p_1^2} : y^2 = x^3 + u.9$, where $E_1/\mathbb{F}_{p_1} : y^2 = x^3 + 9$, $p_1 = 2^{127} - 58309$, u is a non-square in $\mathbb{F}_{p_1^2}$ and $\#E'_1(\mathbb{F}_{p_1^2})$ is a 254-bit prime. Since $p_1 = 3 \pmod{8}$, we represent $\mathbb{F}_{p_1^2}$ as

Table 1. Point multiplication timings (in clock cycles), 64-bit processor

Method	# of cores	AMD Phenom II
$E'_1(\mathbb{F}_{p_1^2})$ 127-bit p , 4GLV+INT, 7pts.	4	91,000
$E'_1(\mathbb{F}_{p_1^2})$ 127-bit p , 4GLV+INT, 7pts.	1	124,000
$E'_1(\mathbb{F}_{p_1^2})$ 127-bit p , wNAF, 7pts.	1	248,000
$E_2(\mathbb{F}_{p_2})$ 256-bit p , 2GLV+INT, 7pts.	2	136,000

$\mathbb{F}_{p_1}[i]$, where $i = \sqrt{-1}$. Let $u = 1 + i$. The two endomorphisms are given by $\Phi(x, y) = (\xi x, y) = \lambda P$ and $\Psi(x, y) = (u^{(1-p)/3}x, u^{(1-p)/2}y) = \mu P$, where $\xi^3 = 1 \pmod{p_1}$. Following Section 5 it can be verified that $\Phi^2 + \Phi + 1 = 0$ and $\Psi^2 + 1 = 0$. Note that a similar curve was also used in [4] but using a different 4-GLV construction. For the case of the standard 2-GLV, we use the curve $E_2/\mathbb{F}_{p_2} : y^2 = x^3 + 2$, where $p_2 = 2^{256} - 11733$ and $\#E'_2(\mathbb{F}_{p_2})$ is a 256-bit prime. The endomorphism is given by $\Phi(x, y)$.

For our experiments we used a 3.0GHz AMD Phenom II X4 940 processor with four cores. The expected timings in terms of clock cycles are displayed in Table 1. As can be seen, closely following our analysis and considering that the use of multiple cores inserts certain penalty, 4-GLV on four cores injects a speed up close to 3x in comparison with a fully sequential version on one core, and supports a computation that runs in 0.67 the time of the standard 2-GLV on two cores.

Although these experimental results correspond to a $j = 0$ curve, we can confidently express that the relative improvement from 2-GLV to 4-GLV on any single GLV curve will be by the same order. This follows theoretically from the minimality of Theorem 1 and the form of Theorem 3, which together imply that $\frac{\text{bitlength of 4-GLV coefficients}}{\text{bitlength of 2-GLV coefficients}} \approx 1/2$ (no r, s involved).

9 Conclusion

We have produced new families of GLV curves, and written all such curves (up to isomorphism) with nontrivial endomorphisms of degree ≤ 3 . We have shown how to generalize the Gallant-Lambert-Vanstone scalar multiplication method by combining it with the Galbraith-Lin-Scott ideas, to perform a proven almost fourfold speedup on GLV curves over \mathbb{F}_{p^2} . We have provided a first explicit bound on such a decomposition using the LLL algorithm. We have then refined this bound using a faster new reduction algorithm, which consists in basically two applications of the extended Euclidean algorithm, one in \mathbb{Z} and the other in $\mathbb{Z}[i]$. This allows us to get a relative improvement (on the same curve) from 2-GLV to 4-GLV *independent* of the curve.

Acknowledgements: We would like to thank Mike Scott for advice on a first version of this work. Also, we would like to thank Diego F. Aranha for his advice on multi-core programming and Joppe Bos for his help on looking for efficient chains for implementing modular inversion.

References

1. H. Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics*. Springer, 1996.
2. S. D. Galbraith, X. Lin, and M. Scott. Endomorphisms for faster elliptic curve cryptography on a large class of curves. In A. Joux, editor, *Proceedings of EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 518–535. Springer, 2009.

3. R. P. Gallant, J. L. Lambert, and S. A. Vanstone. Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms. In J. Kilian, editor, *Advances in Cryptology - Proceedings of CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 190–200. Springer, 2001.
4. Z. Hu, P. Longa, and M. Xu. Implementing 4-dimensional GLV method on GLS elliptic curves with j -invariant 0. *Designs, Codes and Cryptography*, to appear, 2011. Also in Cryptology ePrint Archive, Report 2011/315, <http://eprint.iacr.org/>.
5. A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.
6. P. Longa. *High-speed elliptic curve and pairing-based cryptography*. PhD thesis, University of Waterloo, 2011. <http://hdl.handle.net/10012/5857>.
7. P. Longa. *Elliptic Curve Cryptography at High Speeds*. Talk at the 15th Workshop on Elliptic Curve Cryptography (ECC2011), INRIA, France, 2011. <http://ecc2011.loria.fr/slides/longa.pdf>.
8. P. Longa and A. Miri. New composite operations and precomputation scheme for elliptic curve cryptosystems over prime fields. In Cramer R., editor, *Proceedings of PKC 2008*, volume 4939 of *Lecture Notes in Computer Science*, pages 229–247. Springer, 2008.
9. F. Sica, M. Ciet, and J-J. Quisquater. Analysis of the Gallant-Lambert-Vanstone method based on efficient endomorphisms: Elliptic and hyperelliptic curves. In H. Heys and K. Nyberg, editors, *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 21–36. Springer, 2002.
10. H. M. Stark. Class-numbers of complex quadratic fields. In *Modular functions of one variable, I (Proc. Internat. Summer School, Univ. Antwerp, Antwerp, 1972)*, pages 153–174. Lecture Notes in Mathematics, Vol. 320. Springer, Berlin, 1973.
11. P. Q. Nguyễn and D. Stehlé. Low-dimensional lattice basis reduction revisited. In Duncan A. Buell, editor, *Algorithmic Number Theory, 6th International Symposium, ANTS-VI, Burlington, VT, USA, June 13-18, 2004, Proceedings*, volume 3076 of *Lecture Notes in Computer Science*, pages 338–357. Springer, 2004.
12. Z. Zhou, Z. Hu, M. Xu, and W. Song. Efficient 3-dimensional GLV method for faster point multiplication on some GLS elliptic curves. *Inf. Proc. Lett.*, 77(262):1075–1104, 2010.